IN THE CLAIMS

Please amend claims 1, 10-14, 20, 23, 25, 26, 28, and 30-33 and add claims 34-36 as indicated below.

1. (Currently Amended) A processing system comprising:

a plurality of shadow register sets;

an interrupt vector generator, for generating an exception vector associating with an interrupt handler, when the processing system receives an interrupt;

shadow set mapping logic, coupled to both said plurality of shadow register sets, and said interrupt vector generator, for selecting one of said plurality of shadow register sets to be used by said interrupt handler;

wherein said shadow set mapping logic comprises a plurality of entries, each of which is programmable to associate ~~is programmably provided with a correlation between said~~ exception vectors ~~and said selected~~ with at least one of said plurality of shadow register sets.

2. (Original) The processing system as recited in claim 1 wherein each of said plurality of shadow register sets comprise a plurality of registers that duplicate registers within a general purpose register (GPR) set.

3. (Original) The processing system as recited in claim 1 wherein said correlation comprises a map register that maps said exception vector to said selected one of said plurality of shadow register sets.

4. (Original) The processing system as recited in claim 1 wherein said interrupt vector generator selects a particular one of a plurality of interrupt routines to be used to handle said interrupt.

5. (Original) The processing system as recited in claim 4 wherein said selected particular one of a plurality of interrupt routines is located in a memory at said exception vector.

6. (Original) The processing system as recited in claim 1 wherein said interrupt vector generator selects a particular one of a plurality of interrupt routines to be used to handle said interrupt based on the priority level of said interrupt.

7. (Original) The processing system as recited in claim 1 wherein said shadow set mapping logic further comprises: a plurality of programmable fields, each corresponding to one of a plurality of exception vectors, each of said plurality of fields containing data referencing one of said plurality of shadow register sets.

8. (Original) The processing system as recited in claim 7 wherein each of said plurality of programmable fields comprises a four-bit field.

9. (Original) The processing system as recited in claim 7 wherein said data in each of said plurality of fields corresponds to one of said plurality of shadow register sets.

10. (Currently Amended) The processing system as recited in claim 1 further comprising:

a previous set data register, whose contents indicate which of the plurality of shadow register sets the processing system should use to complete handling of a first interrupt after handling of a second interrupt is complete; and

a current set data register, whose contents indicate which of the plurality of shadow register sets the processing system should use to handle the second interrupt~~ a status register, for storing data corresponding to a current shadow set, and a previous shadow set~~.

11. (Currently Amended) The processing system as recited in claim 10 wherein said ~~status register comprises: a current set data register; and a previous set data register~~ shadow set mapping logic is configured to convey to the current set data register an indication as to which of said plurality of shadow register sets is to be used by an interrupt handler associated with the first interrupt vector.

12. (Currently Amended) The processing system as recited in claim ~~11~~ 10 wherein when said second interrupt is received by the processing system, the contents of said current set data register are moved to said previous set data register.

13. (Currently Amended) The processing system as recited in claim ~~3~~ 10 wherein when said second interrupt has been handled, the contents of said previous set data register are restored into said current set data register.

14. (Currently Amended) A microprocessor for receiving a plurality of interrupts, and for executing a plurality of interrupt routines corresponding to the plurality of interrupts, comprising:

a general purpose register set, having a plurality of registers addressable by instructions;

a plurality of shadow register sets, each having a plurality of registers that are addressable by said instructions; and

shadow set mapping logic, coupled to both said general purpose register set and said plurality of shadow register sets, for selecting one of said plurality of shadow register sets to be addressable by said instructions upon receipt of one of the plurality of interrupts wherein said shadow set mapping logic comprises:

a plurality of programmable fields, each of which is programmable to associate ~~corresponding to one of said plurality of interrupt routines, for storing data indicating which~~ one of said plurality of shadow register sets ~~are to be utilized for said field's corresponding~~ with at least one of said plurality of interrupt routines.

15. (Original) The microprocessor as recited in claim 14, wherein the plurality of interrupts comprise: off-core interrupts from an interrupt controller; and on-core interrupts generated by a core.

16. (Cancelled)

17. (Original) The microprocessor as recited in claim 14 further comprising: a vector generator, coupled to said shadow set mapping logic, for receiving the plurality of interrupts, and for each one of the plurality of interrupts, selecting a corresponding one of the plurality of interrupt routines to be executed.

18. (Original) The microprocessor as recited in claim 17, wherein said vector generator provides an interrupt vector corresponding to said selected one of the plurality of interrupt routines to said shadow set mapping logic.

19. (Original) The microprocessor as recited in claim 18, wherein upon receipt of said interrupt vector, said shadow set mapping logic selects a corresponding one of a plurality of vector fields that contains data indicating which of said plurality of shadow register sets is to be used for said selected one of the plurality of interrupt routines.

20. (Currently Amended) Register Set selection logic within a microprocessor, the microprocessor receiving a plurality of interrupts that are serviced by corresponding ones of a plurality of interrupt routines, the selection logic comprising:

a plurality of register sets, each having a plurality of registers, wherein a first one of said plurality of register sets is directly addressable by instructions;

a vector generator, for receiving the plurality of interrupts, and for generating a plurality of exception vectors, each relating to the corresponding ones of the plurality of interrupt routines; and

mapping logic, coupled to both of said plurality of register sets and said vector generator, for selecting ones of said plurality of register sets for use by said plurality of interrupt routines wherein said mapping logic further comprises a plurality of fields, each <u>of which is programmable to associate</u> ~~corresponding to one of said plurality of~~ exception vectors~~;~~

~~wherein said mapping logic selects said ones~~ with at least one of said plurality of register sets ~~utilizing said exception vectors~~.

21. (Cancelled)

22. (Previously Presented) The register set selection logic as recited in claim 20 wherein each of said plurality of fields contains an indicator that references one of said plurality of register sets.

23. (Currently Amended) A microprocessor having a first register set for use by non-interrupt instructions, and second and third register sets for use by interrupt service routines, the microprocessor comprising:

a vector generator, for generating exception vectors corresponding to the interrupt service routines; and

programmable mapping logic, coupled to said vector generator, comprising a plurality of entries, each of which is programmable to associate exception vectors with either ~~for programmably selecting between~~ the second or the ~~and~~ third register sets for use by the interrupt service routines, based on a value of said exception vectors.

24. (Original) The microprocessor as recited in claim 23 wherein said mapping logic comprises: a first entry ~~field~~ corresponding to a first exception vector; and a second entry ~~field~~ corresponding to a second exception vector.

25. (Currently Amended) The microprocessor as recited in claim 24 further comprising:

a previous set data register, whose contents indicate which of the plurality of shadow register sets the processing system should use to complete handling of a first interrupt after handling of a second interrupt is complete; and

a current set data register, whose contents indicate which of the plurality of shadow register sets the processing system should use to handle the

second interrupt ~~wherein each of said first field and said second field contains data referencing either the second register set or the third register set~~.

26. (Currently Amended) A method within a processing system for utilizing shadow register sets and shadow set mapping logic for programmably mapping interrupts to the shadow register set, the method comprising:

upon receipt of an interrupt, determining which one of a plurality of exception routines should be executed; and

based on the received interrupt, selecting one of a plurality of shadow register sets to be utilized by the one of the plurality of exception routines;

wherein said step of selecting utilizes programmable registers within the shadow set mapping logic, each of said programmable registers ~~that contain~~ being configured to store data which associates exception routines with at least ~~indicating which~~ one of the plurality of shadow register sets ~~is to be used for its interrupt~~.

27. (Original) The method as recited in claim 26 wherein each of the programmable registers corresponds to one of the exception routines.

28. (Currently Amended) A ~~computer program product for use with a computing device, the computer program product comprising:~~

~~a~~ computer readable storage ~~usable~~ medium having computer readable program code embodied in said medium, for causing a microprocessor to be described, said computer readable program code comprising:

first program code for providing a plurality of shadow register sets;

second program code for providing programmable shadow set mapping logic comprising a plurality of entries, each of which is programmable to associate exception vectors with at least one ~~for selecting ones~~ of the plurality of shadow register sets to be utilized by interrupt routines addressed by exception vectors;

third program code for providing a status register, the status register having a current shadow set reference and a previous shadow set reference;

~~wherein the mapping logic contains programmable fields that correspond to the exception vectors;~~ and

wherein when an interrupt routine completes utilization of its shadow register set, contents of the previous shadow set reference are placed into the current shadow set reference.

29. (Cancelled)

30. (Currently Amended) The computer readable storage medium ~~program product~~ as recited in claim 28 further comprising: fourth program code for providing a vector generator, for receiving interrupts and for generating the exception vectors.

31. (Currently Amended) A computer readable storage medium ~~data signal embodied in a transmission medium~~ comprising:

computer-readable program code for providing a microprocessor having a general purpose register set, and a plurality of shadow register sets, said program code comprising:

first program code for providing a vector generator, for receiving interrupts and for generating exception vectors corresponding to each of the received interrupts;

second program code for providing mapping logic, the mapping logic having a plurality of fields, each of which is programmable to associate exception vectors with at least one of the plurality of shadow register sets ~~fields corresponding to one of the generated exception vectors~~; and

third program code for providing shadow register selection logic, for reading the contents of the field corresponding to a generated one of the exception vectors, and selecting one of the plurality of shadow register sets based on the contents of the field.

32. (Currently Amended) The computer readable storage medium ~~data signal~~ as recited in claim 31 wherein the fields are programmable by kernel mode instructions.

33. (Currently Amended) The computer readable storage medium ~~data signal~~ as recited in claim 32 wherein by programming the fields, a particular one of the plurality of shadow register sets is configured to be utilized by a particular exception routine referenced by a particular exception vector.

34. (New) The method as recited in claim 26, further comprising:

storing in a previous set data register, an indication of which of the plurality of shadow register sets the processing system should use to complete handling of a first interrupt after handling of a second interrupt is complete; and

storing in a current set data register, an indication of which of the plurality of shadow register sets the processing system should use to handle the second interrupt.

35. (New) The method as recited in claim 34, further comprising the shadow set mapping logic conveying to the current set data register an indication as to which of said plurality of shadow register sets is to be used to handle the first interrupt.

36. . (New) The method as recited in claim 35, further comprising when said second interrupt is received by the processing system, moving the contents of said current set data register to said previous set data register.